# Measures of Context-Awareness for Self-Organizing Systems

Andrei Olaru, Cristian Gratie, and Adina Magda Florea

Department of Computer Science,
University Politehnica of Bucharest
313 Splaiul Independentei, Bucharest, 060042 Romania
cs@andreiolaru.ro, cgratie@yahoo.com, adina@cs.pub.ro

**Abstract.** The plethora of interconnected devices that surrounds modern people has yet to work together as a whole. An intelligent environment must sense and react to the actions of people, but to that end a large quantity of information must be exchanged throughout the system. Under realistic conditions, it is impossible to control and coordinate the exchange of information in a centralised way. Solving this problem involves key concepts like self-organisation, emergent behaviour and context-awareness. Continuing previous work on self-organising cognitive multi-agent systems for the exchange and management of information, this paper introduces two aspects of context-awareness – pressure and interest – that make the system's emergent behaviour more context-sensitive and, therefore, more adaptive to a changing environment.

## 1 Introduction

People in the modern world are surrounded by a huge number of electronic devices that have different capabilities, different sizes and different performance, all of them interconnected by wireless or wired networks, but not quite working together and cooperating towards the resolution of tasks.

Ambient intelligence is the field that deals with electronic environments that are sensitive and responsive to the presence and actions of people [1]. Ambient intelligence implies, on the one hand, embedded, non-intrusive and personalised interfaces and, on the other hand, an underlying ubiquitous network that links the devices together into one system.

Therefore, what a real-world Ambient Intelligence scenario involves is a huge number of devices that are able to communicate with each other within a certain range. These devices – mobile phones, PDAs, mp3 players, sensors embedded in clothes or in the environment, vehicles, and many more – must provide an interface between the human users and the Ambient Intelligent system, and, through this interface, assist the user in daily activities. Assisting the user is

[1] This paper is based on the paper by the same authors presented in the ACSys 2009 Workshop, 11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania

done by providing information that may be considered interesting, information that is conveyed and aggregated within the system.

From our point of view, one essential issue is what happens *between* the upper level of the human-machine interface and the bottom level of the interconnecting network. The exact manner in which information is transferred between devices is far from having a trivial solution. First, all aspects of one's life taken into account, the quantity of information that transits the system is very large. Second, most of the devices that use the information have reduced storage and processing capacity. As the devices move all the time, together with their owner, they have to act in an autonomous manner. They also need to be proactive and consider a great deal of contextual information. With this requirements, we consider that an agent-oriented approach is the most adequate. Moreover, considering the required flexibility of the system and the number of devices involved, it is obvious that centralised control is not viable, therefore the system must self-organise.

We have addressed the issue of a multi-agent self-organising system for the management of information in our previous work [6, 7]. However, the notion of context representation and context-awareness had not yet been integrated.

Context-awareness implies that the behaviour of an agent (that is assigned to a device) takes into consideration various information that change according to the situation the device and its owner are in. For example, different behaviour is required when the user is shopping and when the user is in an emergency condition. Also, different actions should be taken if a piece of information provided by the rest of the system is considered as being interesting to the user versus if that is not the case.

This paper takes a step forward towards the implementation of context-awareness in a system for the distributed management of information, using agents that are cognitive but have limited storage capacity. In this case, a simple and generic enough representation of context is necessary, one that is easy to process but also useful enough for the functioning of the system. The paper proposes two elements of context representation, namely *pressure* and *interest*. The two elements relate to two aspects of context awareness: the relevance that the source associates to a piece of information (this is the pressure) and the relevance that the receiver associates with the piece of information (this is the interest). Such a system for the emergent management of information, using the two aspects of context awareness, has been designed and implemented

Section 2 is dedicated to related work in the fields of context-awareness, emergence and self-organisation. Section 3 presents the requirements and the general description of the implemented system. Section 4 introduces some aspects of context awareness in the described system and two elements for the representation of context are proposed: pressure and interest. Section 5 describes the design of the system and the results of the experiments are discussed in section 6. The last section draws some conclusions.

## 2 Related Work

Self-organisation and emergent behaviour in multi-agent systems are usually inspired from biological systems and have been studied mostly by using reactive agents [8, 5]. Most of the obtained emergents consist of the organisation of agents in a spacial or space-related structure that groups agents with certain states. The limitation of these systems comes from the use of reactive agents, that, as opposed to cognitive agents, are very simple and therefore cannot lead to complex functions at the level of the system.

Self-organisation has also been studied for systems based on agents having a more elaborate model, but such approaches are less frequent [3, 4] and are used for the distributed resolution of tasks rather then for the exchange and management of information.

Research in context-awareness offers complex solutions even in the field of mobile devices [2], but the representation of context lacks generality and is difficult to implement using really small amounts of memory. This paper aims to offer a solution that requires very low storage and processing capacity. Moreover, the agent's behaviour is tuned in accordance to the context, enabling it to act with increased promptness whenever the situation requires it.

Our previous work [7] described a system for the exchange and management of information. Unlike most studies of self-organising systems, our approach used cognitive agents instead of reactive ones, because of their increased flexibility and possibilities. The interaction between agents lead to emergent properties related to the uniform distribution of data. However, the context was not taken into account.

## 3 System Requirements

The purpose of the system we designed is to manage information in such a way that its users will have interesting or needed information available without the need to know where this information comes from or how it was made available to them. From the user's perspective, two operations are possible: either insert information into the system or request certain information he/she might be interested in. Also, the system might provide the user with information that is potentially interesting to him/her. We will discuss each of these key concepts in the following paragraphs.

The system is implemented as a cognitive multi-agent system. In general, each agent is assigned to a user and is responsible for the management of the data that is of interest to its assigned user.

The different pieces of information (results, announcements and others) will be referred to generically as *data*. Data is characterised by its content, by the size of its content and by context-specific information.

## 4 Aspects of Context Awareness

In previous experiments with the information management system that we have developed [7], two issues were encountered. First, the behaviour of the agents was only regulated by the quantity of plans and messages that an agent had and was not related to the informational content of the agent's knowledge or to the received data. The agent acted the same way in the case of receiving data that needed a quick spreading (e.g. an announcement of emergency or a very important announcement from another user) and in the case of data for which there is no such requirement (e.g. normal information, promotional offers etc). Second, the data was distributed by the same rules, independent of its content or relatedness with any domain. The two issues are both related to the lack of context awareness.

The solution for the first issue is the integration of a measure of urgency associated with the data, representing how quickly the data should be handled and how important it is for it to be passed on. This measure was called *pressure*. Higher pressure means more urgency and the data should spread more and faster throughout the system.High pressure may be associated with very important announcements. Lower pressure is associated with information that is only meant to spread slower, in a more limited area.

Pressure is also associated with requests for data. The higher the pressure, the more urgent the request for data is and the system should react more quickly and provide the necessary data to the requesting user. Lower pressure means that the user does not expect the data to be available immediately and the system is allowed to react less promptly.

The global pressure of all facts in the knowledge base represents the pressure on the agent. As discussed later on, the agent will act differently when it is under pressure and when it is relaxed.

The second issue relates to the definition of *interest*, measured according to some *domains* of interest. In this paper, three generic domains of interest are considered, named $A$, $B$ and $C$.

Each piece of data is associated with a measure of interest, comprising the amount of relatedness between the data and each domain of interest. For example, data that is in domain $A$ but also has a degree of relationship of 0.2 with domain $C$ will have a measure of interest represented as $\langle A : 1.0,\ C : 0.2\rangle$. We will call this *data-interest*.

Each agent has a measure of its interest towards the three domains, calculated and adjusted according to the data that was produced and/or requested by its associated user. We will call this *agent-interest*. Also, each belief that an agent has about some data or some other agent is associated with an unidimensional measure of interest that represents how interesting that fact is to the agent. We will call this *fact-interest*. The representation of facts is discussed later on.

Considering the multi-dimensional (in our generic case, three-dimensional) measures of interest as vectors, the interestingness of a particular belief is calculated as a function of the norm of the difference between the vectors of interest associated with the data and with the agent. The fact-interest is normalised to be in the interval [0, 1].

For example, if an agent interested in domains A (more) and C (less) – agent-interest is $\langle A : 0.9,\ C : 0.3\rangle$ – learns (from another agent) a fact about data D – data-interest $\langle A : 0.1,\ B : 0.2,\ C : 0.9\rangle$ – the resulting fact-interest associated to the fact in the agent's knowledge base will be the equal to $1 - ||\langle 0.9, 0.0, 0.3\rangle - \langle 0.1, 0.2, 0.9\rangle||/\sqrt{3} = 0.41$. The resulting interest will be moderate mostly because the agent is only moderately interested in domain C, that the data is mostly related to.

## 5 System Design

The system is conceived as a two-dimensional space in which cognitive agents are placed and each agent has a number of acquaintances with which it can communicate directly. For easier representation, experiments were carried out using agents placed in a rectangular grid, each agent communicating directly with its 8 neighbours.

### 5.1 Agent Beliefs

Agents are cognitive and implement the Belief-Desire-Intention model. The beliefs of an agent are held in its knowledge base and are represented using three structures: *Data*, *Goal* and *Fact*. A *Data* structure represents information about one piece of data: the content, the size and the interest relative to the domains. A *Goal* structure contains information about an objective, or goal, of an agent. One example of an agent's goal is "need to get data D", represented as $\langle Get, D\rangle$, where $D$ is a *Data* structure. However, what agents hold in their knowledge bases and what they send to each other are *Fact* structures. A *Fact* is a tuple that can have one of the following forms:

$$\langle Agent, Data, pressure, interest\rangle$$

$$\langle Agent, Goal, pressure, interest\rangle$$

$$\langle Agent, Fact, pressure, interest\rangle$$

These three types of facts represent associations between the specified elements. The first type of fact means that "*Agent* has *Data*"; the second – "*Agent* has *Goal* objective"; the third type means that "*Agent* knows *Fact*". All three types contain an indication of *pressure* on the fact. The pressure shows how important this fact is for the *Agent*. In the first and third cases, it shows how quickly the fact should be processed and / or sent to the neighbours. In the second case, it shows how important it is for the agent to fulfill the goal. The interest is the *fact-interest* associated with the fact, as specified in section 4.

It is very important to note the use of the third type of fact, i.e. agents may know facts about what other agents know. This kind of knowledge may also be passed to other agents, if it is of interest. In order for the agent not to be overwhelmed by irrelevant facts about distant agents, the knowledge base is updated by removing the facts that the agent is least interested in.

Another remark about the third type of fact must be made. As this type is recursive, it is obvious that the last nested fact must be of one of the other two types. The knowledge base of an agent must carefully check the contained facts so that no circularity appears.

### 5.2 Agent Goals

The choice of individual agent goals is particularly important for our approach, as the system is meant to exhibit global, emergent properties as a result of local goals and local interaction between agents. As stated before, the fact that the agents only have local goals and a local image of their environment means that they need less computing power and may be deployed on smaller devices.

The global goal of the system is to integrate new information coming from the users and to make it available to other users that need it or might find it interesting. Therefore, at the local level, agents should interact with their neighbours in order to exchange data and collaborate. With this goal in mind, the agents were designed with human-inspired features that would help them share relevant information as well as prevent them from being overwhelmed by facts or by the data itself.

The agents have been given the following goals (no particular order was used):

- fulfill external (human user) requests;
- keep some storage available in case data is injected from the exterior (by the human user);
- share data (according to context);
- get interesting data from other agents;
- help other agents fulfill their goals.

Goals are chosen according to their importance, which is represented by means of *pressure*. In most cases, the pressure is taken from the fact associated with the goal: for external requests and for helping other agents, it is a fact of the form $\langle Agent, Goal\rangle$ that is received from a neighbour or from the exterior; for getting interesting data, it is (possibly nested into other facts) a fact of the form $\langle Agent, Data\rangle$. The goal of keeping storage available has a pressure which varies exponentially with the amount of used capacity over 75%, as it is essential to have some free storage available at any time so that the user can insert new data.

### 5.3 Agent Plans

The plans contain the actions that an agent must perform in order to fulfill its goals. In the system that we have designed, an agent may have several ongoing plans at the same time, and may also have a set of *waiting* plans, i.e. their completion depends on an external event, like knowledge or data expected to come from another agent, as a result of a request.

The plans that an agent may build are formed of several basic actions, like:

- send data to a neighbour, following a request;
- request data from a neighbour, if the agent knows that the neighbour has that data;
- inform a neighbour of a fact that the agent believes relevant to the neighbour. This will be done only if the agent believes that the neighbour does not already know that fact. The fact may express that an agent has certain data, or that an agent has a certain goal;
- discard some data, according to its relevance to the agent's domain of interest and according to the recent frequency of requests (external or not) that have been made for that data.

### 5.4 Agent Behaviour

The behaviour of the agents is fairly normal for a BDI architecture, but it has some particularities that make them suitable for the required application.

The stages that an agent goes through during a step of the system's evolution are:

- **Receive data from and send data to** neighbour agents. Data is transmitted only as a response to previous requests. These operations are simple and need no reasoning, and that is why they are handled separately and before any other decision is made.
- **Revise beliefs**. This is done based on information received from the other agents. Duplicate facts are found and removed. Also, circular facts are identified and only those containing at most one cycle are kept, i.e. it is allowed for agent A to know that agent B knows that agent A knows fact F, but not more. The interest and pressure of new facts are computed. An important thing to note here is that the pressure of received facts is decreased, so that the pressure of a fact diminishes with each step it takes farther from its source.
- **Check ongoing or waiting plans** for completion (was the goal achieved?) and test whether they have become impossible. In the case of completion the plan is discarded and the pressure of the corresponding facts is cancelled. The interest towards the facts, on the other hand, remains constant.
- **Make plans**. Take the goal with the highest current importance (pressure). If there is no plan for it, make a plan composed of the actions needed to be taken and put it in the list of ongoing plans.
- **Execute plans**. Take the plan associated with the most important goal and execute its next action. Each executed action reduces the pressure of the associated goal with some amount. If there are no more actions to be performed, move the plan to the list of waiting plans. It will be checked for completion in the next cycle.

– **Fade memory** of all facts in the knowledge base. Pressure of all facts is faded. Facts in which the agent has no interest may be discarded. This step is necessary in order to avoid overwhelming the agent with useless facts and too old concerns.
– **Revise pressure and interest**. The pressure on the agent is recalculated according to the facts in its knowledge base, as a weighted mean of the pressures of the individual facts, giving more importance to high-pressure facts. The interest is updated as well, according to the pieces of data that the agent holds and according to its recent activity (the resulting interest is also calculated as a mean).

As it is also the case for natural systems, the behaviour of the agent is greatly influenced by the pressure that presses upon it. Besides the instantaneous pressure, the agent's state is also described by a lower and a higher limit for the pressure. If the pressure is between the two limits, activity is considered normal. If pressure is lower than the lower limit, the agent is considered "relaxed". If the pressure is above the higher limit, the agent is considered "stressed".

The more "stressed" the agent is, the more it will focus on completing its most pressing plans. New knowledge and data acquisition will be reduced to a minimum.

The lower and higher pressure limits are not fixed, and they are adjusted in time. If the pressure on the agent continues to be high for a long time, the two limits will rise and the agent will start to consider its condition as a "normal" one.

## 6 Experiment

There were many experiments carried out in the study of context-aware emergent behaviour. The scenario presented in this paper is generic, focusing on showing how the proposed context-aware measures influence the spreading of information through the system.

### 6.1 Scenario

There were two directions that were followed in the experiments: the study of the evolution of pressure in the system and the study of the system's behaviour in the context of facts from different areas of interest.

The behaviour of complex systems is extremely difficult to observe in simple graphical representations and the study of the system's behaviour means, most times, the minute observation of the activity log of each agent. This is why a simple scenario was used, one that makes the evolution of agent states clearer.

This is also the reason why some visualisation tools have been developed. The graphs that were used are of three types:

– data-fact distribution – associated with a certain piece of data; the graphic is a two-dimensional grid representation of the system where the color of a

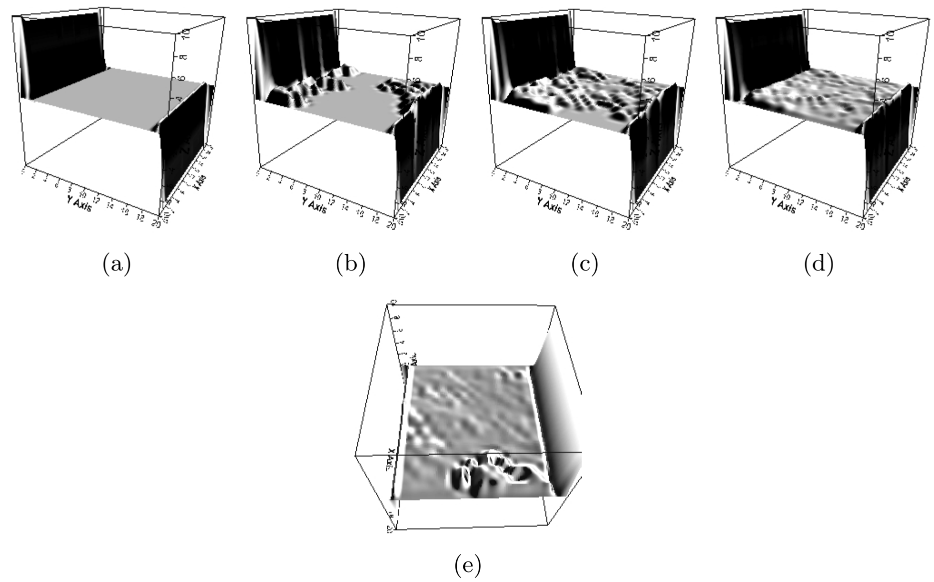

**Fig. 1.** Pressure distributions: (a) initial state (step 0); (b) immediately after inserting 3 new pieces of data into the system (step 5); (c), (d) stabilisation of pressure across the whole system (steps 10 and 20); (e) a "wave" of pressure at the insertion of a new piece of data (step 35).

cell shows the existence of facts regarding the data in the knowledge base of the agent in that cell (e.g. any graphic from Figure 2 (a), (b) or (c)).

- pressure distribution – a three-dimensional surface showing the amount of pressure (z-axis) on each agent in the system (e.g. Figure 1 (a) - (e)). Please note that, for technical reasons, the pressure distributions in Figure 1 (a)-(d) are shown with the top row of the system's grid closer to the viewer.
- interest distribution – a two-dimensional grid representation of the system where the colour of a cell shows the amount of interest that an agent has for a certain domain (e.g. Figure 3 (c)). The colour of each cell is generated by direct conversion of the agent-interest (which has three components in the interval [0,1]) into an RGB code.

The scenario that is used for exemplification contains two major steps. There are three domains of interest – A, B and C. Initially, there is no data stored in the system and agents have no facts in their knowledge bases. During the first step of the system's evolution, three pieces of data – D1, D2 and D3 – are inserted into the system through the agents in three corners of the system's grid. Data D1 – inserted in the bottom-left corner – is relevant to domain A; data D2 – inserted in the top-right corner – is relevant to domain B and slightly relevant to domain C; data D3 – inserted in the bottom-right corner – is relevant only to domain B. After the system stabilises, at step 30 two more pieces of data –

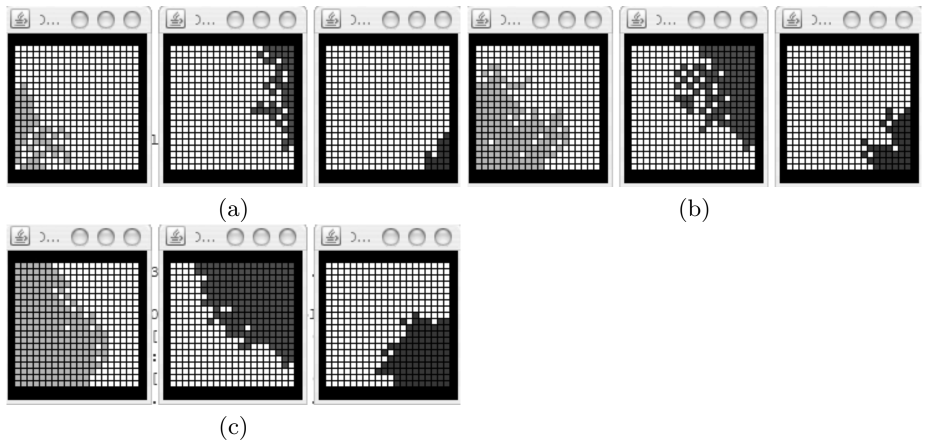


**Fig. 2.** (a), (b), (c) Distribution of facts regarding data D1, D2 and D3 at steps 5, 10 and 20 (pressure at those steps is represented in Figure 1 (b), (c) and (d) )

D4 and D5 – are inserted simultaneously into two agents close to each other, on the bottom row of the system's grid. Data D4 is relevant to domain B; data D5 is relevant to domain A.

### 6.2 Results and Discussion

Relevant results that have been obtained after experimentation are displayed in Figures 1 and 3, also considering Figure 2 for reference on the system's evolution in the first phase (steps 0 to 30).

**Pressure** indicates the importance and urgency of a piece of data or fact, and, in the case of agents, the need for quick reaction. This makes facts with higher fact-pressure spread quicker and over larger areas. Observe that in Figure 3 (d), (e) and (f) facts about data D4 spread more as a result of D4 having higher pressure than D5.

High agent-pressure makes agents focus only on their most important concerns. Pressure on an agent decreases as the important plans are being executed and as time passes. Observe in Figure 1 the evolution of the pressure on the agents in the system. At first there is no pressure (Figure 1 (a)); after the insertion of new data, great pressure is put on the agents that must spread facts about that data (Figure 1 (b)). Note that, in areas where all agents know the facts, pressure becomes more uniform. In time, and with the completion of goals, pressure decreases and becomes uniform throughout the whole system (Figure 1 (d)). When new data is inserted, pressure rises again (Figure 1 (e)). Observe the "wave" of pressure. Behind the "wave", pressure is high, but more uniform, and decreasing. On the edge of the "wave", pressure is non uniform, as there are still some agents that haven't received the facts yet (also refer to Figure 3 (d)).

Pressure relates to urgency. On the other hand, **interest** controls the direction of information exchange and the area of spread, not in terms of size, but in

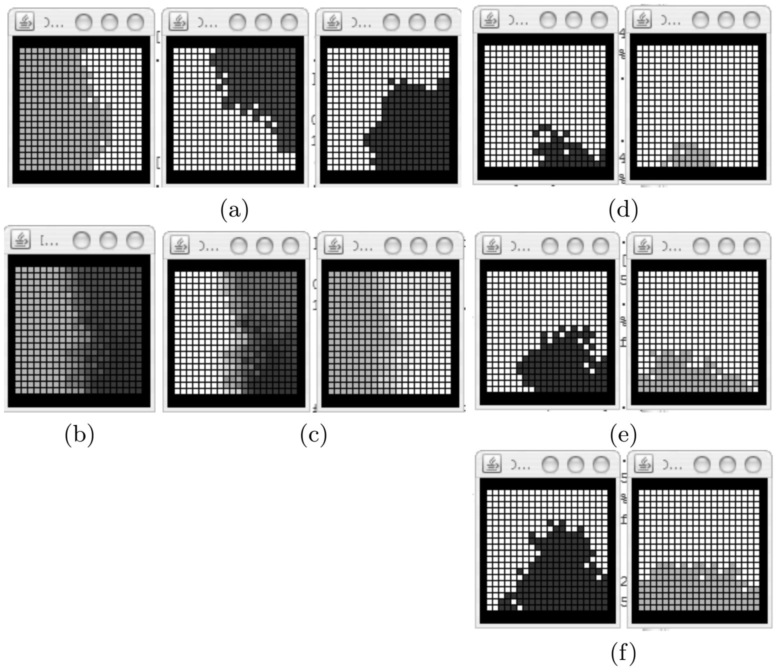

**Fig. 3.** (a) Distribution of facts regarding pieces of data 1 (relevant to domain A), 2 (relevant to domain B and slightly relevant to domain C) and 3 (relevant to domain B); (b) distribution of the interest of agents in all domains; (c) distribution of the interest of agents in domains A and B. Shades of gray represent the actual measure of the interest towards the domain; (d), (e), (f) Evolution of facts about data 4 (relevant to domain B) and data 5 (relevant to domain A), at steps 39, 47 and 54.

terms of the previous experience of agents. That is, agents that have exchanged more facts related to a certain domain of interest will become more interested in that domain. Positive feedback will lead to the formation of groups of agents interested in a certain domain.

A good example of this behaviour is shown in Figure 3. The facts regarding the first pieces of data inserted into the system – D1, D2 and D3 – have spread into three different areas of the system (Figure 3 (a)). Their spread stopped partly because pressure decreased with their advance and with time, and partly because agents that have specialised in one domain will be less interested in the other domains. Note that the areas of spread for facts regarding data D2 and D3 (both in the domain of interest B) have a larger intersection than the areas of D1 and D2 $\cup$ D3, that are from different domains of interest. In this situation, two more pieces of data are inserted into the system: D4 (related to domain B) and D5 (related to domain A). From the beginning, it is easy to observe that their spread is guided by the interest of the agents (Figure 3 (d), (e)): data D4 spreads preferentially on the right side of the grid, and data D5 spreads more on the left side. Data D5 also spreads to the right, but long after D4.

The experiments have shown that exchanging information based on context leads to good results, even if the representation of context is very primitive.

## 7 Conclusion

One of the essential elements required for the implementation of ambient intelligence is a decentralised, self-organising exchange of information between devices of reduced storage and processing capacity.

This paper describes a multi-agent system for information exchange that includes the key concept of context awareness, by using a simple but effective representation of context, comprised of two elements: pressure and interest. The system has been tested on a scenario involving a large number of agents and experiments have shown that the system, as a whole, is indeed considering context in its behaviour.

The paper does not refer to a specific application, as the system and its behaviour have been kept as generic as possible, trying to deal with the general problem of context-aware information exchange.

## References


1. Augusto, J., McCullagh, P.: Ambient intelligence: Concepts and applications. Computer Science and Information Systems/ComSIS **4**(1), 1–26 (2007)
2. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Tech. rep., Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College (2000)
3. Gleizes, M., Camps, V., Glize, P.: A theory of emergent computation based on cooperative self-organization for adaptive artificial systems. In: Fourth European Congress of Systems Science (1999)
4. Hales, D., Edmonds, B.: Evolving social rationality for MAS using "tags". Proceedings of the second international joint conference on Autonomous agents and multiagent systems pp. 497–503 (2003)
5. Mano, J.P., Bourjot, C., Leopardo, G., Glize, P.: Bio-inspired mechanisms for artificial self-organised systems. Special Issue: Hot Topics in European Agent Research II Guest Editors: Andrea Omicini **30**, 55–62 (2006)
6. Olaru, A., Florea, A.M.: Emergence in cognitive multi-agent systems. Proc. of CSCS17, 17th International Conference on Control Systems and Computer Science, Bucharest, Romania pp. 515–522 (2009)
7. Olaru, A., Gratie, C., Florea, A.M.: Emergent properties for data distribution in a cognitive mas. Proc. of the 3rd International Symposium on Intelligent Distributed Computing - IDC 2009, October 13-14, Ayia Napa, Cyprus (2009). (in print)
8. Serugendo, G.D.M., Gleizes, M.P., Karageorgos., A.: Self-organization and emergence in MAS: An overview. Informatica **30**(1), 45–54 (2006)